1. *MAP interpretation of regularized empirical loss minimization.* We have seen that some (unregularized) empirical loss minimization problems can be interpreted as maximum likelihood estimation (MLE) if we choose certain parametric form for the conditional probability $p(y|\boldsymbol{x};\boldsymbol{\theta})$. Assuming the data samples are i.i.d., MLE of $p(y|\boldsymbol{x};\boldsymbol{\theta})$ is equivalent to

$$\underset{\boldsymbol{\theta}}{\text{minimize}} \quad \sum_{i=1}^{n} -\log p(y_i|\boldsymbol{x}_i;\boldsymbol{\theta}).$$

After some trivial transformations, we can recover some supervised learning models such as least squares regression and logistic classification.

Some statisticians, who call themselves Bayesians, believe that we should treat $\boldsymbol{\theta}$ as random as well, and impose probability distributions on them. In this case, the probability that we really care about is $p(\boldsymbol{\theta}|Y,\boldsymbol{X})$, the conditional probability of $\boldsymbol{\theta}$ given data $\boldsymbol{X} = \{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n\}$ and $Y = \{y_1,\ldots,y_n\}$. According to Bayes rule,

$$p(\boldsymbol{\theta}|Y,\boldsymbol{X}) = \frac{p(Y|\boldsymbol{X},\boldsymbol{\theta})p(\boldsymbol{\theta}|\boldsymbol{X})}{p(Y|\boldsymbol{X})}.$$

Furthermore, it is common to assume that $\boldsymbol{\theta}$ is independent of $\boldsymbol{X}$ and $(\boldsymbol{x}_i, y_i)$ are i.i.d. conditioned on $\boldsymbol{\theta}$, leading to

$$p(\boldsymbol{\theta}|Y,\boldsymbol{X}) = \frac{p(\boldsymbol{\theta})\prod_{i=1}^{n} p(y_i|\boldsymbol{x}_i,\boldsymbol{\theta})}{p(Y|\boldsymbol{X})}.$$

Here, $p(\boldsymbol{\theta})$ is called the prior (*a priori* in Latin), $p(y|\boldsymbol{x},\boldsymbol{\theta})$ is called the likelihood, and $p(\boldsymbol{\theta}|Y,\boldsymbol{X})$ is called the posterior (*a posteriori* in Latin).

Depending on the definition of the prior and the likelihood, the denominator $p(Y|\boldsymbol{X})$ may be very hard to evaluate. Instead, we can try to find a point estimate $\boldsymbol{\theta}$ that maximizes the posterior probability, which is called maximum *a posteriori* (MAP), since the denominator does not depend on $\boldsymbol{\theta}$ and can be omitted in maximization. This is equivalent to

$$\underset{\boldsymbol{\theta}}{\text{minimize}} \quad \sum_{i=1}^{n} -\log p(y_i|\boldsymbol{x}_i,\boldsymbol{\theta}) - \log p(\boldsymbol{\theta}).$$

For each of the following cases, given an explicit MAP formulation for estimating $\boldsymbol{\theta}$. Find their relationship to the corresponding regularized empirical loss minimization problems. Specifically, give an exact expression for the regularization parameter λ in terms of the prior and likelihood distributions.

(a) $p(y|\boldsymbol{x},\boldsymbol{\theta}) \sim \mathcal{N}(\boldsymbol{\phi}^\top\boldsymbol{\theta}, \sigma^2)$ and $p(\boldsymbol{\theta}) \sim \mathcal{N}(0, \sigma_0^2\boldsymbol{I})$;

(b) $p(y|\boldsymbol{x},\boldsymbol{\theta}) \sim \mathcal{N}(\boldsymbol{\phi}^\top\boldsymbol{\theta}, \sigma^2)$ and $p(\boldsymbol{\theta})$ follows a multivariate Laplacian distribution:

$$p(\boldsymbol{\theta}) = \prod_{j=1}^{m} \frac{1}{2a}\exp\left(-\frac{|\theta_j|}{a}\right);$$

(c) $p(y|\boldsymbol{x},\boldsymbol{\theta}) = \Pr[yu \geq 0]$ where $y = \pm 1$, $p(u|\boldsymbol{x},\boldsymbol{\theta}) \sim \mathcal{N}(\boldsymbol{\phi}^\top\boldsymbol{\theta}, \sigma^2)$ and $p(\boldsymbol{\theta}) \sim \mathcal{N}(0, \sigma_0^2\boldsymbol{I})$;

(d) $p(y|\boldsymbol{x},\boldsymbol{\theta}) = 1/(1+\exp(-y\boldsymbol{\phi}^\top\boldsymbol{\theta}))$ where $y = \pm 1$ and $p(\boldsymbol{\theta})$ follows a multivariate Laplacian distribution as in (b).

Ans.1-

(a) $P(y|x,\theta) \sim N(\phi^T\theta, \sigma^2)$

and $P(\theta) \sim N(0, \sigma_0^2 I)$

We know that,

$$-\log P(y_i|x_i,\theta) = -\left(-\frac{1}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}(\phi_i^T\theta - y_i)^2\right)$$

$$= \frac{1}{2}\log(2\pi\sigma^2) + \frac{1}{2\sigma^2}(\phi_i^T\theta - y_i)^2$$

↳ equation ①

and,

$P(\theta) \sim N(0, \sigma_0^2 I)$

$$-\log P(\theta) = \frac{1}{2}\log|\sigma_0^2 I| + \frac{I^{-1}}{2\sigma_0^2}(\theta^T\theta)$$

$$= \frac{1}{2}\log|\sigma_0^2 I| + \frac{I^{-1}}{2\sigma_0^2}\|\theta\|^2$$

↳ equation ②

Since MAP is dependent on θ hence we can ignore constants and our objective function can be written or by combining equation (1) and equation (2).

$$\underset{\theta}{\text{minimize}} \left[\frac{1}{2\sigma^2} \sum_{i=1}^{n} (\phi_i^T \theta - y_i)^2 + \frac{\Sigma^{-1}}{2\sigma_0^2} \|\theta\|^2 \right]$$

$$\underset{\theta}{\text{minimize}} \frac{1}{2\sigma^2} \left[\sum_{i=1}^{n} (\phi_i^T \theta - y_i)^2 + \frac{\sigma^2}{\sigma_0^2} \Sigma^{-1} \|\theta\|^2 \right]$$

Then our function becomes =

$$\underset{\theta}{\text{minimize}} \left[\|\phi\theta - Y\|^2 + \frac{\sigma^2 \Sigma^{-1}}{\sigma_0^2} \|\theta\|^2 \right]$$

This equation is of the form

$$L(\theta) + \lambda \gamma(\theta)$$

where $\lambda = \frac{\sigma^2}{\sigma_0^2} \Sigma^{-1}$

Here $L(\theta) = \|\phi\theta - y\|^2$

which is a least squared loss function.

$r(\theta) = \|\theta\|^2$, it is L2 regularizer

and $\lambda = \frac{\sigma^2}{\sigma_0^2} I^{-1}$

(b) = Given ⇒

$$p(y|x,\theta) \sim \mathcal{N}(\phi^T\theta, \sigma^2)$$

and $p(\theta)$ follows a multivariate laplacian distribution:

$$p(\theta) = \prod_{j=1}^{m} \frac{1}{2a} \exp\left(-\frac{|\theta_j|}{a}\right)$$

Then we can write = ⌐ equation ①

$$-\log p(y|x,\theta) = \frac{1}{2}\log(2\pi\sigma^2) + \frac{1}{2\sigma^2}(\phi_i^T\theta - y_i)^2$$

also we can write =

$$-\log p(\theta) = -\log\left(\prod_{j=1}^{m} \frac{1}{2a} \exp\left(\frac{-|\theta_j|}{a}\right)\right)$$

$$= \sum_{j=1}^{m} \log 2a + \frac{|\theta_j|}{a}$$

$$= m\log 2a + \sum_{j=1}^{m} \frac{|\theta_j|}{a} \rightarrow \text{equation ②}$$

Combining Equation ① and Equation ② we can form the objective function =

$$\underset{\theta}{\text{minimize}} \left[\sum_{i=1}^{n} \left(\frac{\log(2\pi\sigma^2)}{2} + \frac{1}{2\sigma^2}(\phi_i^T\theta - y_i)^2 \right) + n\log 2a + \sum_{j=1}^{m} \frac{|\theta_j|}{a} \right]$$

Since minimizer is θ we can ignore the constants and rewrite our objective function as =

$$\underset{\theta}{\text{minimize}} \left[\sum_{i=1}^{n} \frac{(\phi_i^T\theta - y_i)^2}{2\sigma^2} + \sum_{j=1}^{m} \frac{|\theta_j|}{a} \right]$$

$$= \min_{\theta} \frac{1}{2\sigma^2} \left[\sum_{i=1}^{n} (\phi_i^T\theta - y_i)^2 + \frac{2\sigma^2}{a} \sum_{j=1}^{m} |\theta_j| \right]$$

$$= \min_{\theta} \left[\|\phi\theta - y\|^2 + \frac{2\sigma^2}{a} \|\theta\|_1 \right]$$

This equation is of the form

$$L(\theta) + \lambda r(\theta)$$

where $\lambda = \frac{2\sigma^2}{a}$

In that case

$L(\theta)$ = least Square loss

$r(\theta)$ = L2 regularizer.

(c) $= P(y|x,\theta) = Pr[yu \geq 0]$

where $y = \pm 1$, $P(u|x,\theta) \sim N(\phi^T\theta, \sigma^2)$

and $P(\theta) \sim N(0, \sigma_0^2 I)$

$$P(y|x,\theta) = Pr[yu \geq 0]$$

$$= \frac{1}{\sqrt{2\pi}} \int_0^{\infty} \exp\left(-\frac{(u - y\phi^T\theta)^2}{2}\right) du$$

$$= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{y\phi^T\theta} \exp\left(-\frac{u^2}{2}\right) du$$

$$= \phi(y\phi^T\theta)$$

Then

$$-\log P(y|x,\theta) = -\log(\phi(y\phi^T\theta))$$

↳ equation ①

$$P(\theta) \sim N(0, \sigma_0^2 I)$$

$$-\log P(\theta) = \frac{\theta^T \theta}{2\sigma_0^2 I} + \frac{1}{2}\log 2\pi\sigma_0^2 I$$

↳ Equation (2)

Combining Equation (1) and Equation (2) to form our objective function Since minimizer is θ we can ignore the constants Hence final objective function is =

$$\underset{\theta}{\text{Minimize}} \left[-\log\left(\phi(y\phi^T\theta)\right) + \frac{\|\theta^2\|_2}{2\sigma_0^2 I} \right]$$

form of Equation $L(\theta) + \lambda\, r(\theta)$

Here $\lambda = \frac{1}{2\sigma_0^2}$

In order to find θ where this function minimizer. we have to take partial derivative of the function and put it equal to 0.

$$\frac{\partial}{\partial \theta}\left(-\log\left(\phi\left(y\phi^T\theta\right)\right) + \frac{\partial}{\partial \theta}\frac{\|\theta\|_2^2}{2\sigma_0^2} = 0\right.$$

$$\Rightarrow \frac{-1}{\phi(y\phi^T\theta)}\left(\phi y \phi^T\right) + \frac{\|\theta\|_2^2\left(\theta_1+\theta_2\cdots\theta_n\right)}{\sigma_0^2} = 0$$

This does not gives any solution.

Hence There is no closed solution for Probit classifier.

(d) $= P(y|x,\theta) = \dfrac{1}{(1+\exp(-y\phi^T\theta))}$

Here $y = \pm 1$

and $P(\theta) = \prod_{j=1}^{m} \frac{1}{2a} \exp\left(-\frac{|\theta_j|}{a}\right)$

we can write =

$$-\log P(y|x,\theta) = -\log\left(\frac{1}{1+\exp(-y\phi^T\theta)}\right)$$

$$= \log(1+\exp(-y\phi^T\theta))$$

$\because [-\log \cdot \frac{1}{x} = \log x]$ ↳ equation ①

also we can write =

$$-\log P(\theta) = -\log\left(\prod_{j=1}^{m} \frac{1}{2a} \exp\left(\frac{-|\theta_j|}{a}\right)\right)$$

$$= \sum_{j=1}^{m} \log 2a + \frac{|\theta_j|}{a}$$

$$= m \log 2a + \frac{\|\theta\|}{a}$$ → equation ②

Combining Equation ① and ② to form the objective function ⇒

$$\underset{\theta}{\text{Minimize}} \left[\log\left(1+\exp(-y\phi^T\theta)\right) + m\log 2a + \frac{\|\theta\|}{a} \right]$$

Minimizer is θ and hence we can ignore the constants.

The equation is of form

$$L(\theta) + \lambda\gamma(\theta)$$

where $\lambda = \frac{1}{a}$

In order to find optimal θ. we will have to find partial derivative of function with respect to θ and equate it to 0.

Then we obtain,

$$\sum_{i=1}^{n} \frac{1}{1+\exp(-y\phi^T\theta)} \cdot e^{-y\phi^T\theta} \cdot -y\phi^T + \frac{M}{a} = 0$$

But by using this process we cant obtain θ because logistic regression does not have a closed solution MLE.

Hence in order to obtain θ we have to use gradient descent and then we can obtain approximations for θ.

2. *Nonexpansiveness of proximal operators.* In this problem we show that for a convex function f (not necessarily differentiable), its proximal operator is nonexpansive, i.e.,

$$\|\operatorname{Prox}_f(\boldsymbol{\theta}_1) - \operatorname{Prox}_f(\boldsymbol{\theta}_2)\| \leq \|\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2\|,$$

where

$$\operatorname{Prox}_f(\boldsymbol{\theta}_1) = \arg\min_{\boldsymbol{\theta}} f(\boldsymbol{\theta}) + \frac{1}{2}\|\boldsymbol{\theta} - \boldsymbol{\theta}_1\|^2,$$

with the following steps:

(a) Show that

$$\boldsymbol{\theta}_1 - \operatorname{Prox}_f(\boldsymbol{\theta}_1) \in \partial f(\operatorname{Prox}_f(\boldsymbol{\theta}_1)).$$

(b) Show that if $\boldsymbol{g}_1 \in \partial f(\boldsymbol{\theta}_1)$ and $\boldsymbol{g}_2 \in \partial f(\boldsymbol{\theta}_2)$, then

$$(\boldsymbol{g}_1 - \boldsymbol{g}_2)^\top(\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2) \geq 0.$$

Hint. By definition, if $\boldsymbol{g}_1$ is a subgradient of $f(\boldsymbol{\theta}_1)$, then for all $\boldsymbol{\theta}$

$$f(\boldsymbol{\theta}) \geq f(\boldsymbol{\theta}_1) + \boldsymbol{g}_1^\top(\boldsymbol{\theta} - \boldsymbol{\theta}_1).$$

(c) Use the previous two results to show the *firm nonexpansiveness*

$$(\operatorname{Prox}_f(\boldsymbol{\theta}_1) - \operatorname{Prox}_f(\boldsymbol{\theta}_2))^\top(\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2) \geq \|\operatorname{Prox}_f(\boldsymbol{\theta}_1) - \operatorname{Prox}_f(\boldsymbol{\theta}_2)\|^2.$$

(d) Apply the Cauchy-Schwartz inequality to obtain the nonexpansiveness property.

Ans:3 =

(a) = Show that,

$$\theta_1 - \text{Prox}_f(\theta_1) \in \partial f(\text{Prox}_f(\theta_1))$$

we are also given that, → equation (1)

$$\text{Prox}_f(\theta_1) = \arg\min_{\theta} f(\theta) + \frac{1}{2}\|\theta - \theta_1\|^2$$

In order to obtain Proximal mapping for θ_1 we need to minimize the right hand side with respect to θ.

That means θ is the minimizer of the problem.

Hence By definition we can write =

$$\theta = \text{Prox}_f(\theta_1)$$

By using "fermats optimality condition" and "sum rule of Sub differential calculus" we can minimize the right hand side of equation (1) by differentiating it and putting the solution equivalent to 0. in order to calculate local minima.

Differentiating R.H.S. $\Rightarrow \arg\min_{\theta} f(\theta) + \frac{1}{2}\|\theta - \theta_1\|^2$

we obtain =

$$\partial f(\theta) + \frac{1}{2} \cdot 2 (\theta - \theta_1)(1 - 0) \in 0$$

$$\Rightarrow \partial f(\theta) + \theta - \theta_1 \in 0$$

$$\Rightarrow \quad \theta_1 - \theta \in \partial f(\theta)$$

Since $\theta = Prox_f(\theta_1)$

we can put θ in above equation

Hence we obtain=

$$\theta_1 - Prox_f(\theta_1) \in \partial f\left(Prox_f(\theta_1)\right)$$

Hence Proved

(b)= given, $g_1 \in \partial f(\theta_1)$

$g_2 \in \partial f(\theta_2)$

Then show that,

$$(g_1 - g_2)^T(\theta_1 - \theta_2) \geq 0$$

also we know that =

① = $f(\theta) \geq f(\theta_1) + g_1^T(\theta - \theta_1)$

② = $f(\theta) \geq f(\theta_2) + g_2^T(\theta - \theta_2)$

Then we can write =

$$f(\theta_1) \geq f(\theta_2) + g_2^T (\theta_1 - \theta_2)$$

(Putting $\theta = \theta_1$ in equation ②)

$$f(\theta_2) \geq f(\theta_1) + g_1^T (\theta_2 - \theta_1)$$

(Putting $\theta = \theta_2$ in equation ①)

Adding both the equations above ⇒

$$f(\theta_1) + f(\theta_2) \geq f(\theta_1) + f(\theta_2) + g_1^T(\theta_2 - \theta_1) + g_2^T(\theta_1 - \theta_2)$$

$$\Rightarrow \quad 0 \geq g_1^T(\theta_2 - \theta_1) + g_2^T(\theta_1 - \theta_2)$$

$$\Rightarrow \quad 0 \geq -g_1^T(\theta_1 - \theta_2) + g_2^T(\theta_1 - \theta_2)$$

$$\Rightarrow \quad 0 \geq (\theta_1 - \theta_2)(g_2^T - g_1^T)$$

multiplying both sides by -1. we get,

$$\Rightarrow \quad 0 \leq (\theta_1 - \theta_2)(g_1^T - g_2^T)$$

$$\Rightarrow 0 \leq (\theta_1 - \theta_2)(g_1 - g_2)^T$$

$$\Rightarrow (\theta_1 - \theta_2)(g_1 - g_2)^T \geq 0$$

Hence Proved.

(c) = from part (a) we have $\Rightarrow$ → equation ①

$$\theta_1 - \text{Prox}_f(\theta_1) \in \partial f(\text{Prox}_f(\theta_1))$$

from part (b) we have $\Rightarrow$ → equation ②

$$(\theta_1 - \theta_2)(g_1 - g_2)^T \geq 0$$

Putting $g_1 = \theta_1 - \text{Prox}_f(\theta_1)$

$g_2 = \theta_2 - \text{Prox}_f(\theta_2)$

Putting these values in equation ② $\Rightarrow$

$$(\theta_1 - \theta_2)\left(\theta_1 - \text{Prox}_f(\theta_1) - \left(\theta_2 - \text{Prox}_f(\theta_2)\right)\right)^T \geq 0$$

$$(\theta_1-\theta_2)\left((\theta_1-\theta_2)-\left(prox_f(\theta_1)-prox_f(\theta_2)\right)\right)^T \geq 0$$

$$\Rightarrow (\theta_1-\theta_2)^T(\theta_1-\theta_2)-(prox_f(\theta_1)-prox_f(\theta_2))^T(\theta_1-\theta_2) \geq 0$$

θ is the minimizer hence we can write,

$$\Rightarrow \left(prox_f(\theta_1)-prox_f(\theta_2)\right)^T(\theta_1-\theta_2) \geq \left(prox_f(\theta_1)-prox_f(\theta_2)\right)^T \left(prox_f(\theta_1)-prox_f(\theta_2)\right)$$

$$\Rightarrow \left(prox_f(\theta_1)-prox_f(\theta_2)\right)^T(\theta_1-\theta_2) \geq \left\| prox_f(\theta_1)-prox_f(\theta_2)\right\|^2$$

Hence proved

(a) = $a^T b \leq \|a\| \|b\|$

Hence in the equation =

$$\left(\text{Prox}_f(\theta_1) - \text{Prox}_f(\theta_2)\right)^T (\theta_1 - \theta_2)$$

$$\geq \|\text{Prox}_f(\theta_1) - \text{Prox}_f(\theta_2)\|^2$$

we can write,

$$\|\text{Prox}_f(\theta_1) - \text{Prox}_f(\theta_2)\| \; \|\theta_1 - \theta_2\|$$

$$\geq \|\text{Prox}_f(\theta_1) - \text{Prox}_f(\theta_2)\|^2$$

$$\Rightarrow \|\theta_1 - \theta_2\| \geq \|\text{Prox}_f(\theta_1) - \text{Prox}_f(\theta_2)\|$$

$$\Rightarrow \|\text{Prox}_f(\theta_1) - \text{Prox}_f(\theta_2)\| \leq \|\theta_1 - \theta_2\|$$

Hence proved.

3. *Hand-written digits classification.* The MNIST data set is a famous data set for multi-class classification, which can be downloaded here `http://yann.lecun.com/exdb/mnist/`. In this problem you will design a SGD algorithm for multi-class support vector machine with group-sparse regularization that solves the following optimization problem

$$\underset{\boldsymbol{\theta}_1,\ldots,\boldsymbol{\theta}_k}{\text{minimize}} \quad \frac{1}{n}\sum_{i=1}^{n}\max_{c}(\boldsymbol{x}_i^\top\boldsymbol{\theta}_c - \boldsymbol{x}_i^\top\boldsymbol{\theta}_{y_i} + 1_{y_i\neq c}) + \lambda\sum_{j=1}^{m}\sqrt{\sum_{c=1}^{k}\theta_{jc}^2}.$$

Here we simply assume that the features are the image pixels themselves (we even ignore the constant 1 here).

(a) Derive the stochastic proximal subgradient algorithm for solving it. For simplicity, you can assume that there is only one term that reaches the maximum value in $\max_c(\boldsymbol{\phi}^\top\boldsymbol{\theta}_c - \boldsymbol{\phi}_i^\top\boldsymbol{\theta}_{y_i} + 1_{y_i\neq c})$ throughout the iterations. At iteration t, you can simply denote the step size as $\gamma^{(t)}$.

(b) Implement the algorithm in your favorite programming language.

(c) Run the algorithm with $\lambda = 10, 1, 0.1, 0.01$ and diminishing step size $\gamma^{(t)} = 1/t$, and run the algorithm for 10^6 iterations. At every 1000 iteration, evaluate the prediction accuracy on the test set and plot the progress on a figure.

(d) For the solution of each λ value, show a black and white figure for the pixels that are being used to make the predictions. Is it true that a large λ leads to a more sparse solution?

Ans.3 =

(a) = In order to minimize the objective function of the form ⇒

$$\underset{\theta}{\text{minimize}} \quad L(\theta) + \lambda r(\theta)$$

Here $L(\theta) = \frac{1}{n}\sum_{i=1}^{n}\max(x_i^T\theta_c - x_i^T\theta_{y_i} + 1_{y_i\neq c})$

This is our loss function.

and $r(\theta)$ = regularizer

$$r(\theta) = \sum_{j=1}^{m} \sqrt{\sum_{c=1}^{k} \theta_{jc}^2}$$

We will be using Proximal gradient descent in order to minimize the objective function over θ and obtain optimized θ values.

In order to obtain θ values from Proximal gradient descent we have to do following. =

Proximal gradient descent is given as =

$$\theta^{(t+1)} = \text{Prox}_{\gamma,\lambda r}\left(\theta^t - \gamma \nabla L(\theta^t)\right)$$

$$= \arg\min_{\theta} \frac{1}{2} \| \theta - \theta^t + \gamma \nabla L(\theta^t) \|^2 + \lambda \gamma r(\theta)$$

In order to obtain Proximal gradient we have to obtain gradient of loss function first. i.e. $L(\theta)$

Steps =

① = initially assume parameter (θ) values.

② = obtain gradient of $L(\theta)$

③ = in $L(\theta)$ = input assumed parameter values.

and data from one random image from the test set.

④ = calculate $\theta^{(t+1)}$ for current iteration using Proximal mapping.

⑤ = This will be continued in each iteration till value of θ converges.

(b) = I have implemented the algorithm in python using keras and tennorflow backend.

Algorithm has following steps =

(1) = Obtain MNIST dataset available with keras library.

(2) = importing dataset (training, test) in the program

(3) = converting the image data [28 X 28] from matrix form to vector form (784 X 1)

(4) = assuming parameter values initially

(5) = obtaining loss function

(6) = obtaining objective function.

(7) = Start iterations and calculate proximal gradient descent with $\lambda = 0.5$ and $\gamma = 0.01$

(8) = after every 1000 iterations using obtained θ matrix, calculate prediction on test set and calculate the accuracy.

(9) = Accuracy is printed using a function called Accuracy checker.

The Program is working & it prints accuracy at every 1000 iteration.